INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Multi-task Discriminative Training of Hybrid DNN-TVM Model for Speaker Verification with Noisy and Far-Field Speech

Arindam Jati, Raghuveer Peri, Monisankha Pal, Tae Jin Park, Naveen Kumar, Ruchir Travadi, Panayiotis Georgiou, Shrikanth Narayanan

University of Southern California, Los Angeles, CA, USA

{jati, rperi, mp_323, taejinpa, komathnk, travadi}@usc.edu, {georgiou, shri}@sipi.usc.edu

Abstract

The paper aims to address the task of speaker verification with single-channel, noisy and far-field speech by learning an embedding or feature representation that is invariant to different acoustic environments. We approach from two different directions. First, we adopt a newly proposed discriminative model that hybridizes Deep Neural Network (DNN) and Total Variability Model (TVM) with the goal of integrating their strengths. DNN helps learning a unique variable length representation of the feature sequence while TVM accumulates them into a fixed dimensional vector. Second, we propose a multi-task training scheme with cross entropy and triplet losses in order to obtain good classification performance as well as distinctive speaker embeddings. The multi-task training is applied on both the DNN-TVM model and state-of-the-art x-vector system. The results on the development and evaluation sets of the *VOiCES challenge* reveal that the proposed multi-task training helps improving models that are solely based on cross entropy, and it works better with DNN-TVM architecture than x-vector for the current task. Moreover, the multi-task models tend to show complementary relationship with cross entropy models, and thus improved performance is observed after fusion.

Index Terms: Speaker verification, deep neural networks, total variability model, multi-task training

1. Introduction

The performance of a Speaker Verification (SV) system can deteriorate due to mismatch between training, enrollment and test environments [1, 2]. Data inconsistencies may arise due to channel conditions, noise, background speakers, and microphone placement (near- *vs.* far-field speech) and associated reverberation [1, 3]. One way to tackle the problem is to obtain a speaker representation or embedding [4] that is robust and invariant to different channel and noise conditions. The present work focuses on that approach.

"The VOiCES from a Distance Challenge 2019" [5] has been organized to benchmark state-of-the-art technologies for speaker verification from *single channel* far-field speech in noisy conditions. The evaluation data has been curated from the VOiCES corpus [6], which includes speech data recorded under challenging acoustic environments. On the other hand the training data for the *fixed condition* [5] consists of three "in the wild" datasets, and is not guaranteed to be from similar acoustic conditions (more details in Section 3.1). This paves a way to develop robust speaker embedding systems and evaluate them on realistic far-field speech with natural reverberation [5].

Some of the earlier works [7, 8] for single channel far-field speaker verification focused on designing robust features. Avila *et al.* [9] analyzed performance degradation of classical GMM-UBM [10] and i-vector [11] systems in far-field condition, and proposed multi-condition training with different reverberation levels to address the problem. Similarly, a multi-condition approach was adopted in [12] for training a Gaussian PLDA model in i-vector space. Snyder *et al.* [13] introduced x-vectors which employed different types of artificial augmentation to train a robust speaker embedding using a Time Delay Neural Network- (TDNN) based speaker classification model [14]. Nandwana *et al.* [2] analyzed the performance of the x-vector and i-vector systems for far-field noisy SV task on SRI distant speech collect [2] and VOiCES [6] datasets. X-vectors were shown to have superior performance.

In the present work, we try to address the noisy and far-field SV task from two different angles: potentially find a better model to transform a variable length utterance into a fixed dimensional embedding, and employ a loss function that directly works on the embedding space to reduce intra-speaker distance and increase inter-speaker distance irrespective of channel conditions. The main contributions of this work are the following:

1. We employ a newly proposed [15] hybrid discriminative DNN-TVM system which leverages the strength of both systems. Specifically, it exploits the strength of DNN to project the input feature into a distinctive sequence of vectors, and utilizes TVM to obtain a fixed dimensional embedding.
2. We implement a multi-task training scheme with cross entropy and triplet [16] losses to circumvent the deficiencies of training with only cross entropy loss as observed in computer vision domain [16]. To the best of our knowledge, exploration of this multi-task training has not been done in the past for speaker recognition.

2. Methodology

Total Variability Modeling (TVM) and i-vectors have been the state-of-the-art in speaker recognition for a long time. They were originally proposed as unsupervised generative latent variable model [11]. But, when a large amount of labeled data is available, generative models can have inferior performance compared to discriminative models having very high degrees of freedom. The recent success of x-vectors [13] is an example of that. Moreover, the Gaussian Mixture Model (GMM) assumption on the features might put more constraints on TVM than the distribution-free nature of DNN models. The motivations in developing a discriminative multi-task hybrid DNN-TVM are:

1. It removes the GMM assumption on feature vectors as employed in conventional TVM, and provides a distribution-free formulation [17].
2. TDNNs [18, 14, 13] are found to be good in exploiting temporal context, and training them is generally faster than training recurrent neural networks. We can utilize

http://dx.doi.org/10.21437/Interspeech.2019-3010

the strength of TDNN models and deploy them as a feature transformer.

3. The global statistics pooling of conventional x-vector model can be replaced by a TVM model, and the hybrid model can be trained end-to-end with any discriminative objective function *e.g.,* cross entropy loss.
4. Moreover, such systems can be trained on multiple discriminative tasks instead of only speaker classification, as long as the tasks are complementary to each other.

2.1. Total Variability Model (TVM)

Let us denote an utterance, $\mathbf{X}$ of length T as

$$\mathbf{X} = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{T-1}\} \tag{1}$$

where, $\mathbf{x}_t \in \mathbb{R}^D$ is a feature vector at time t. In TVM [11], $\mathbf{X}$ is represented by a speaker- and channel-*dependent* GMM mean supervector, $\mathbf{M} \in \mathbb{R}^{CD}$. Here, C is the number of Gaussian components in the GMM. $\mathbf{M}$ can be expressed as

$$\mathbf{M} = \mathbf{m} + \mathbf{T}\mathbf{w} \tag{2}$$

where, $\mathbf{m}$ is the speaker- and channel-*independent* GMM mean supervector of the Universal Background Model (UBM), $\mathbf{T} \in \mathbb{R}^{CD \times K}$ is a low rank rectangular total variability matrix, and $\mathbf{w} \in \mathbb{R}^K$ is the i-vector for that utterance. $\mathbf{w}$ has standard normal prior distribution:

$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{3}$$

Conventionally, TVM is set up as a Maximum Likelihood Estimation (MLE) problem and trained by Expectation Maximization (EM) algorithm.

2.2. TDNN and x-vectors

The model for x-vectors [13] is comprised of TDNN at the lower layers followed global statistics pooling. The TDNN layers transform the input utterance $\mathbf{X}$ into a sequence of vectors:

$$\mathbf{G} = \mathbf{h}(\mathbf{X}) = \{\mathbf{g}_0, \mathbf{g}_1, \ldots, \mathbf{g}_{T'-1}\} \tag{4}$$

Then the pooling layer computes statistics (mean and standard deviation) of $\mathbf{G}$, and concatenates them to generate a fixed dimensional vector. It is then projected on an embedding layer:

$$\mathbf{w}_{\text{x-vector}} = \mathbf{A}\left[\mathbb{E}^T(\mathbf{G}) \quad | \quad \mathbb{S}^T(\mathbf{G})\right]^T + \mathbf{b} = \mathbf{f}(\mathbf{X}) \tag{5}$$

where $\mathbb{E}$ and $\mathbb{S}$ denote sample mean and standard deviation respectively. $\mathbf{f}(.)$ is a trainable nonlinear function, and denotes the part of the DNN model from input to the embedding layer. The embedding, $\mathbf{w}_{\text{x-vector}}$ is fed to a shallow classifier network with softmax outputs. The full model is trained end-to-end with cross entropy loss. At test time, $\mathbf{w}_{\text{x-vector}}$ is used for scoring.

2.3. Discriminative DNN-TVM system

2.3.1. Distribution-free TVM formulation

Travadi *et al.* [17] showed that the GMM assumption on features can be removed from the TVM formulation if Baum-Welch statistics of the features are used as variables in the model likelihood function instead of the features themselves. The zeroth and first order Baum-Welch statistics are given by:

$$N_c = \sum_{t=0}^{T-1} \gamma_{tc} \quad \text{and} \quad \mathbf{F_c} = \frac{1}{N_c} \sum_{t=0}^{T-1} \gamma_{tc}\mathbf{x}_t \tag{6}$$

Here, $\gamma_{tc} = p(c|\mathbf{x}_t)$ is the posterior probability of the t^{th} feature vector to belong to the c^{th} Gaussian component in the GMM. It was shown in [17] that the statistics $\mathbf{F_c}$ asymptotically follow normal distribution irrespective of the feature distribution. More importantly, the posterior probability, γ_{tc} is *not* limited to be from a GMM distribution, but can be any positive function, $\gamma_{tc} = \Gamma_c(\mathbf{x}_t)$ s.t. $\Gamma_c : \mathbb{R}^D \mapsto [0, \infty)$.

2.3.2. Discriminative TVM training

The MLE of TVM assumes the data (features) were generated from a GMM distribution and tries to find model parameters that can best explain the data. The distribution-free TVM formulation has no assumption on how the features were generated, and thus best viewed as a trainable function from the features to the i-vector. In this case, the TVM parameters can be trained using cross entropy loss with the help of speaker labels. The discriminative training of TVM was proposed in [19] where the authors adopted numerical optimization algorithm to train the parameters, and found superior performance than generative TVM.

2.3.3. Hybrid DNN-TVM model

The hybrid architecture comprises of the initial TDNN layers as in x-vector system. After the frame-level processing is done by the TDNN layers, a trainable mapping, $\Gamma'_c(.)$ is applied on the transformed sequence $\mathbf{G}$ (equation 4) to produce posterior probability for every $\mathbf{g}_t$:

$$\gamma_{tc} = \Gamma'_c(\mathbf{g}_t) \tag{7}$$

The overall transformation of a feature vector $\mathbf{x}_t$ to the posterior is given by (refer to Section 2.2 and 2.3.1):

$$\gamma_{tc} = \Gamma_c(\mathbf{x}_t) = \Gamma'_c(\mathbf{h}(\mathbf{x}_t)) = \Gamma'_c(\mathbf{g}_t) \tag{8}$$

Instead of computing the global statistics of $\mathbf{G}$ as done in x-vector, we compute the Baum-Welch statistics of $\mathbf{G}$ using equation 6. Based on the foundation of Section 2.3.1, these statistics can be used in a TVM formulation.

Intuitively, the posterior γ_{tc} denotes the probability of a feature vector, $\mathbf{x}_t$ to belong to a certain region in the feature space, and $\mathbf{F_c}$ denotes the *local mean* of the features in that region. We concatenate multiple local means $\{\mathbf{F_c}\}_{c=1}^C$ to create a *local mean supervector*. Global mean pooling (as done in x-vector) can be regarded as a special case of this formulation, where $C = 1$ and $\Gamma'_c(.) = 1$.

Finally, we project the local mean supervector to an embedding layer, $\mathbf{w}_{\text{hybrid}}$ through an affine transform. Adopting same convention as equation 5, we can have:

$$\mathbf{w}_{\text{hybrid}} = \mathbf{A}\left[\mathbf{F}_1^T \quad | \quad \mathbf{F}_2^T \quad |\ldots| \quad \mathbf{F}_C^T\right]^T + \mathbf{b} = \mathbf{f}(\mathbf{X}) \tag{9}$$

Similar to x-vector, $\mathbf{w}_{\text{hybrid}}$ then goes to a shallow classifier network. The whole network can be trained using cross entropy loss as done in [15].

2.4. Triplet loss

Theoretically, any discriminative loss function can be used to train the DNN-TVM model (and also for x-vector system as will be used for comparison in Section 4.1). [15] employed standard cross entropy loss, L_{CE} which tries to increase the softmax posterior probabilities for all samples. The drawback of cross entropy loss is that it does not explicitly focus on reducing intra-class variance. On the contrary, triplet loss [16] directly works on the embedding space and tries to bring samples from same class closer than samples from two different classes.

Assume $\mathbf{X}_i^a$ and $\mathbf{X}_i^p$ are two utterances from the same speaker, and are denoted as anchor and positive utterances respectively. $\mathbf{X}_i^n$ is called a negative utterance, and it belongs to a different speaker than anchor and positive. Then the tuple $(\mathbf{X}_i^a, \mathbf{X}_i^p, \mathbf{X}_i^n) \in \mathcal{T}$ is denoted as a triplet, where $\mathcal{T}$ is the set of all triplets. The DNN should ideally find a mapping $\mathbf{f}(.)$ such that,

$$||\mathbf{f}(\mathbf{X}_i^a) - \mathbf{f}(\mathbf{X}_i^p)||_2^2 + \alpha < ||\mathbf{f}(\mathbf{X}_i^a) - \mathbf{f}(\mathbf{X}_i^n)||_2^2 \quad \forall (\mathbf{X}_i^a, \mathbf{X}_i^p, \mathbf{X}_i^n) \in \mathcal{T} \tag{10}$$

where, α is positive margin parameter. This is achieved by minimizing the triplet loss given by:

$$L_{\text{Triplet}} = \frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|}\max\left(||\mathbf{f}(\mathbf{X}_i^a) - \mathbf{f}(\mathbf{X}_i^p)||_2^2 + \alpha - ||\mathbf{f}(\mathbf{X}_i^a) - \mathbf{f}(\mathbf{X}_i^n)||_2^2, 0\right) \quad (11)$$

The mapping $\mathbf{f}(\mathbf{X})$ is generally the DNN that transforms $\mathbf{X}$ into an embedding. For x-vector and hybrid systems, $\mathbf{f}(\mathbf{X})$ is basically $\mathbf{w}_{\text{x-vector}}$ and $\mathbf{w}_{\text{hybrid}}$ respectively (equations 5 and 9).

2.5. Multi-task training

Triplet loss has been successfully applied for speaker verification in past works [20, 21]. Inspired from the success in computer vision domain [22], we incorporate triplet loss along with the cross entropy loss, and train the network in multi-task setting. Note that the triplet loss is directly computed on the embedding, while the cross entropy loss encounters a shallow classifier network after the embedding layer. Our hypothesis is that the multi-task training would force the network to produce both correct classification and distinctive embeddings, and these two tasks would be complementary. More formally, the joint loss function is given by:

$$L = \lambda L_{\text{CE}} + (1-\lambda)L_{\text{Triplet}} \quad (12)$$

where, $0 < \lambda < 1$ helps weighing different losses.

3. Experimental Setting

3.1. Datasets and features

The training datasets for the "fixed condition" [5] in the VOiCES challenge are Voxceleb 1 & 2 [23], and Speakers In The Wild (SITW) [24]. Both the datasets have 16KHz single channel audio. First, we remove 60 overlapped (between Voxceleb and SITW) speakers from the SITW dataset before training. Then we remove speakers that have less than 10 utterances. This results in 7537 unique speakers and $\sim$ 1.3M utterances for training [1].

The official VOiCES development set has around 16K utterances of noisy and far-field speech from 196 speakers, and $\sim$ 4M trials for scoring. The evaluation set has around 11K utterances and $\sim$ 3.6M trials from 100 speakers.

We extract 20 dimensional MFCC features with 25ms window and 10ms shift using Kaldi toolkit [25]. Energy based VAD is applied. The features are mean normalized with a moving window of maximum 3s length. Delta and delta-delta features are concatenated with original MFCC to produce 60 dimensional features for training.

3.2. Data Augmentation

First, every utterance in the training data is augmented with three different types of noise:

- **Television**: To create a simulated environment similar to background television sounds, we first extract the audio from four publicly available video datasets: AVA-ActiveSpeaker dataset [26], advertisement dataset [27], and two compilation videos for TV shows "Friends" and "How I met your mother" available in YouTube. For every training utterance, a random segment from one of these four datasets is picked and added with the original signal with 13-20dB SNR.
- **Babble**: Similar to [13], three to seven speakers are randomly chosen from the above four datasets, summed and added to the original speech at 13-20dB SNR.
- **Music**: A single music file is randomly sampled from the MUSAN music dataset [28] and added to each utterance as described in [13].

Then we reverberate the clean and above three copies with a Room Impulse Response (RIR) randomly sampled from a pool of 60K RIRs [29]. The final training data also keeps the original clean utterance. Due to the large size of the augmented dataset (5 times the original, so $\sim$ 6.5M utterances), and lack of time and resources, **we could only train on a subset of the data**. The subset is created by sampling a maximum of 300 utterances (after augmentation, so a maximum of 60 clean utterances) from each speaker, thus limiting to $\sim$ 2.1M utterances.

3.3. System parameters

3.3.1. i-vector and x-vector baselines

A GMM with 2048 components and full covariance matrix is trained for the UBM. 400 dimensional i-vector extraction system is built on the longest 100K utterances following Kaldi's Voxceleb v1 recipe [2].

The x-vector model is as described in [13], but we utilize our augmented data (Section 3.2) instead of the default augmentation recipe described in [13].

3.3.2. Hybrid DNN-TVM

The hybrid DNN-TVM model is composed of: **TDNN** $\rightarrow$ **Dense** $\rightarrow$ **Dense** $\rightarrow$ **Dense** $\rightarrow$ **Softmax** $\rightarrow$ **TVM Layer** $\rightarrow$ **Classifier**. The TDNN is part of the x-vector model up to '*frame5*' [13]. The dense layers have 1024 hidden units. The softmax layer has 1000 units (analogous to a GMM with 1000 Gaussian components). Three dense layers and the subsequent softmax layer together implement $\Gamma_c'(.)$ as described in Section 2.3.3. The TVM layer computes the Baum-Welch statistics and the local mean supervector. The classifier network is same as [13]. It consists of two 512 dimensional dense layers, and the final softmax layer for classification.

3.3.3. Triplet and multi-task

Mining good triplets is very crucial in triplet learning. Easy triplets might stagnate the training, while very hard triplets might make the training unstable and result in collapsed model [16]. Moreover, the total number of all triplets grows exponentially with the number of samples. We adopt online (in-batch) semi-hard triplet mining [16] because it provides much faster training than offline mining, and is found to address the training issues mentioned above.

For the multi-task loss, we choose $\lambda = 0.8$ in equation 12. As explained in [22], giving more weight to cross entropy loss is slightly beneficial. For our experiment, we observed it facilitated faster convergence at the beginning of the training.

3.4. LDA and PLDA scoring

For all systems (listed in Section 4.1), LDA has been applied to reduce the embedding dimension. The LDA dimension is tuned on the VOiCES development set. For i-vector and multi-task models, 200 dimensional LDA was found to be optimal, while for x-vector 150 dimensional LDA gave the best performance (similar to [13]). After LDA, the embeddings are length normalized. Finally, PLDA training and scoring are performed following the conventions in [13].

[1] We held out the test part of Voxceleb 1 as an internal clean validation set to monitor possible degradation of model on clean speech.

[2] The submitted i-vector system is trained on clean data. Training on the augmented data was not finished before system submission deadline.

Table 1: *Performance of different systems on VOiCES development and evaluation sets. The check marks (✓) indicate the systems that were submitted for official evaluation. The last two rows indicate systems after score fusion.*

	VOiCES development set				VOiCES evaluation set			
System	**minDCF**	**actDCF**	**EER (%)**	**CLLR**	**minDCF**	**actDCF**	**EER (%)**	**CLLR**
1. i-vector clean	0.64	8.94	7.02	0.79	0.99	29.58	31.89	3.51
2. x-vector kaldi (✓)	0.39	4.85	3.42	0.43	0.62	4.35	7.54	0.58
3. x-vector native	0.59	7.08	5.23	0.62	0.86	6.54	11.74	0.76
4. DNN-TVM	0.62	10.58	5.95	0.87	0.89	9.45	12.09	0.93
5. x-vector multi-task	0.41	5.54	3.53	0.49	0.68	5.41	8.05	0.65
6. DNN-TVM multi-task	0.40	6.77	3.83	0.59	0.64	5.37	8.05	0.64
(2+5+6) (✓)	0.36	0.36	**3.18**	**0.13**	**0.60**	**0.62**	**7.29**	**0.45**
(1+2+6) (✓)	**0.35**	**0.35**	3.29	**0.13**	0.67	0.67	8.78	0.53

3.5. Fusion of multiple systems

Given the wide diversity between train and test scenarios, it is often not possible to come up with a single good system for speaker verification. Therefore, to maximize benefit from the complementary merits of different SV systems, we employ a weighted-sum log-likelihood score-fusion strategy. System fusion works well if the fused subsystems are similar in nature, however not identical, and also have complementary characteristics [30]. In this work, we have fused four SV systems: i-vector, x-vector, and the multi-task version of x-vector and DNN-TVM models. Fusion weights and a bias term to perform linear score-fusion are determined with BOSARIS[3] toolkit on development data. The toolkit uses a fast unconstrained convex optimization algorithm based on a quasi-Newton method to train the logistic regression fusion [31].

4. Results and Discussions

4.1. Performance of individual systems

Four metrics are reported here as shown in Table 1. The primary and secondary metrics are minDCF and CLLR respectively, as defined in [5].

The first six rows of Table 1 show the performances of the six individual systems on VOiCES development and evaluation sets. The i-vector system trained on the clean dataset is outperformed by all DNN-based systems. Moreover, the performance of i-vectors degrade severely when we move from development to evaluation set.

Two x-vector systems are developed as shown in rows 2 and 3 of Table 1. The *"x-vector kaldi"* is developed using kaldi's [25] x-vector training recipe, while *"x-vector native"* is our own implementation of x-vector architecture in Keras [32]. The initiative to re-implement the x-vector system arises from the requirements of easy extension and modification of the model using widely used deep learning tools like Keras, and a fair comparison with DNN-TVM model which is implemented using the same tool[4].

Row 4 shows the performance of the hybrid DNN-TVM model. For this application, both implementations of x-vector perform better than the hybrid model in development as well as evaluation datasets.

Rows 5 and 6 show performances of the multi-task models (implemented in Keras). In the evaluation set, the x-vector multi-task training gets a relative improvement of 21% in minDCF from its cross entropy counterpart ("x-vector native"). Similarly, in the evaluation set, the DNN-TVM multi-task is ahead of cross entropy based DNN-TVM by 28% in terms of minDCF. Interestingly, multi-task training of DNN-TVM model works better than the multi-task version of x-vector model. In the evaluation set, DNN-TVM multi-task has a relative advantage of 6% than x-vector multi-task in terms of minDCF.

4.2. Performance of fused systems

The last two rows of Table 1 report the two best performing fused systems. We can see that both the fused systems achieve improvements over the individual systems in the development set, but only system (2+5+6) does so for the evaluation set.

Inclusion of the clean i-vector model in system (1+2+6) deteriorates the performance in evaluation set although promising performance was observed in the development set. We believe this comes from poor generalization of the clean i-vector system as discussed in Section 4.1.

In terms of minDCF, the best fused system, (2+5+6) is about 8% and 3% better than "x-vector kaldi" in development and evaluation sets respectively.

5. Conclusion and Future Directions

The paper focused on speaker verification with noisy and far-field speech. We tried to address the problem by employing a recently proposed hybrid DNN-TVM model. Moreover, a multi-task training scheme was proposed for both state-of-the-art x-vector system and the hybrid model. The multi-task approach jointly optimized cross entropy loss and triplet based similarity loss to achieve both good categorization and distinctive embeddings.

The results on VOiCES development and evaluation sets showed that the multi-task models (both x-vector and DNN-TVM) are better than our native implementations of cross entropy based x-vector and DNN-TVM models. Moreover, they provided complimentary information when combined together with the x-vector system, and thus obtained improved performance compared to individual systems. The multi-task training was found to work better on the DNN-TVM model than the x-vector model for this far-field SV task.

In the future, we plan to do an intensive analysis of the performance gap between ours and kaldi's x-vector implementations, because the gap might also create potential degradation in our DNN-TVM system and its multi-task version. We also plan to train the systems on the full 6.5M augmented utterances, which we could not do due to lack of resource and time. This might fulfill the data hungry needs of DNN and potentially improve the performance.

6. Acknowledgements

The support of the sponsors of this research are gratefully acknowledged.

[3] https://sites.google.com/site/bosaristoolkit/

[4] We noticed a gap in performance between two x-vector implementations possibly because of various custom optimizations done in Kaldi.

7. References

[1] J. H. Hansen and T. Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal processing magazine*, vol. 32, no. 6, pp. 74–99, 2015.

[2] M. K. Nandwana, J. van Hout, M. McLaren, A. Stauffer, C. Richey, A. Lawson, and M. Graciarena, "Robust speaker recognition from distant speech under real reverberant environments using speaker embeddings," in *Proc. Interspeech*, 2018.

[3] Q. Jin, T. Schultz, and A. Waibel, "Far-field speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2023–2032, 2007.

[4] M. McLaren, D. Castan, M. K. Nandwana, L. Ferrer, and E. Yilmaz, "How to train your speaker embeddings extractor," in *Odyssey 2018 The Speaker and Language Recognition Workshop*. ISCA, 2018, pp. 327–334.

[5] M. K. Nandwana, J. Van Hout, M. McLaren, C. Richey, A. Lawson, and M. A. Barrios, "The voices from a distance challenge 2019 evaluation plan," *arXiv preprint arXiv:1902.10828*, 2019.

[6] C. Richey, M. Barrios, Z. Armstrong, C. Bartels, H. Franco, M. Graciarena, A. Lawson, M. K. Nandwana, A. Stauffer, J. van Hout, P. Gamble, J. Hetherly, C. Stephenson, and K. Ni, "Voices obscured in complex environmental settings (voices) corpus," in *Interspeech*, 2018, pp. 1566–1570.

[7] T. H. Falk and W.-Y. Chan, "Modulation spectral features for robust far-field speaker identification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 1, pp. 90–100, 2010.

[8] S. O. Sadjadi, T. Hasan, and J. H. Hansen, "Mean hilbert envelope coefficients (mhec) for robust speaker recognition," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[9] A. R. Avila, M. Sarria-Paja, F. J. Fraga, D. O'Shaughnessy, and T. H. Falk, "Improving the performance of far-field speaker verification using multi-condition training: The case of gmm-ubm and i-vector systems," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[10] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[11] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[12] D. Garcia-Romero, X. Zhou, and C. Y. Espy-Wilson, "Multicondition training of gaussian plda models in i-vector space for noise and reverberation robust speaker recognition," in *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2012, pp. 4257–4260.

[13] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[14] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification." in *Interspeech*, 2017, pp. 999–1003.

[15] R. Travadi and S. Narayanan, "Total variability layer in deep neural network embeddings for speaker verification," *Accepted in IEEE Signal Processing Letters*, 2019.

[16] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[17] R. Travadi and S. S. Narayanan, "A distribution free formulation of the total variability model." in *INTERSPEECH*, 2017, pp. 1576–1580.

[18] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[19] O. Glembek, L. Burget, N. Brümmer, O. Plchot, and P. Matějka, "Discriminatively trained i-vector extractor for speaker verification," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[20] C. Zhang and K. Koishida, "End-to-end text-independent speaker verification with triplet loss on short utterances." in *Interspeech*, 2017, pp. 1487–1491.

[21] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304*, 2017.

[22] X. Zhang, F. Zhou, Y. Lin, and S. Zhang, "Embedding label structures for fine-grained feature representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1114–1123.

[23] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.

[24] M. McLaren, L. Ferrer, D. Castan, and A. Lawson, "The speakers in the wild (sitw) speaker recognition database." in *Interspeech*, 2016, pp. 818–822.

[25] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," IEEE Signal Processing Society, Tech. Rep., 2011.

[26] J. Roth, S. Chaudhuri, O. Klejch, R. Marvin, A. Gallagher, L. Kaver, S. Ramaswamy, A. Stopczynski, C. Schmid, Z. Xi *et al.*, "Ava-activespeaker: An audio-visual dataset for active speaker detection," *arXiv preprint arXiv:1901.01342*, 2019.

[27] Z. Hussain, M. Zhang, X. Zhang, K. Ye, C. Thomas, Z. Agha, N. Ong, and A. Kovashka, "Automatic understanding of image and video advertisements," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1705–1715.

[28] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[29] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5220–5224.

[30] O. Glembek, F. Grézl, M. Karafiát, D. A. van Leeuwen, P. Matejka, P. Schwarz, and A. Strasheim, "Fusion of heterogeneous speaker recognition systems in the stbu submission for the nist speaker recognition evaluation 2006," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, 2007.

[31] C.-J. Lin, R. C. Weng, and S. S. Keerthi, "Trust region newton methods for large-scale logistic regression," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 561–568.

[32] F. Chollet *et al.*, "Keras," https://keras.io, 2015.